Kilobot: A Robotic Module for Demonstrating Behaviors in a Large Scale (\(2^{10}\) Units) Collective

Kilobot: A Robotic Module for Demonstrating Behaviors in a Large Scale (2^{10} units) Collective

Michael Rubenstein, Radhika Nagpal

***Abstract*— A collective of robots can together complete a task that is beyond the capabilities of any of its individual robots. One property of a robotic collective that allows it to complete such a task is the shape of the collective. This paper presents Kilobot, a simple modular robot designed to work in a collective to self-assemble and self-heal that collective's shape.**

In previous work, an algorithm is given that allows a simulated collective of robots to self-assemble and self-heal a desired shape, keeping the shape sized proportional to the number of robots in the collective. In this abstract, the current work of producing a robotic collective that can demonstrate that algorithm is presented.

I. INTRODUCTION

It is possible for a group of robots, called a collective, to complete a goal that cannot be completed by any of the individual robots. One way to accomplish this is if the collective forms a specific shape. For example, imagine a single SWARM-BOT [1] reaches a canyon-like obstacle with a goal on the other side. By itself, a single SWARM-BOT is not capable of crossing the canyon to reach the goal on the other side. However, if the SWARM-BOT joins a collective of other SWARM-BOTs, and forms a collective shaped like a bridge, the collective's shape enables it to cross the canyon and reach the goal. In another example, a single Superbot robot [2] needs to locomode as far as it can until its battery pack empties. As a solitary Superbot, it can only travel 200 meters until the battery is empty. If this single Superbot can form a collective with five other Superbot robots in the shape of a wheel, then it can move over 1000 meters until its battery pack depletes. In this case, the shape of the Superbot collective enables the collective to travel five times as far as any single Superbot can travel.

In previous work [3,4] a distributed control method called S-DASH was presented which enables a collective of robots to form a given shape at a scale proportional to the total number of robots. If the shape of the collective is damaged, for example by removing some robots, then S-DASH will reform the shape at a new, smaller scale, proportional to the new number of robots. A demonstration of this behavior in a simulated collective is shown in Fig. 1.

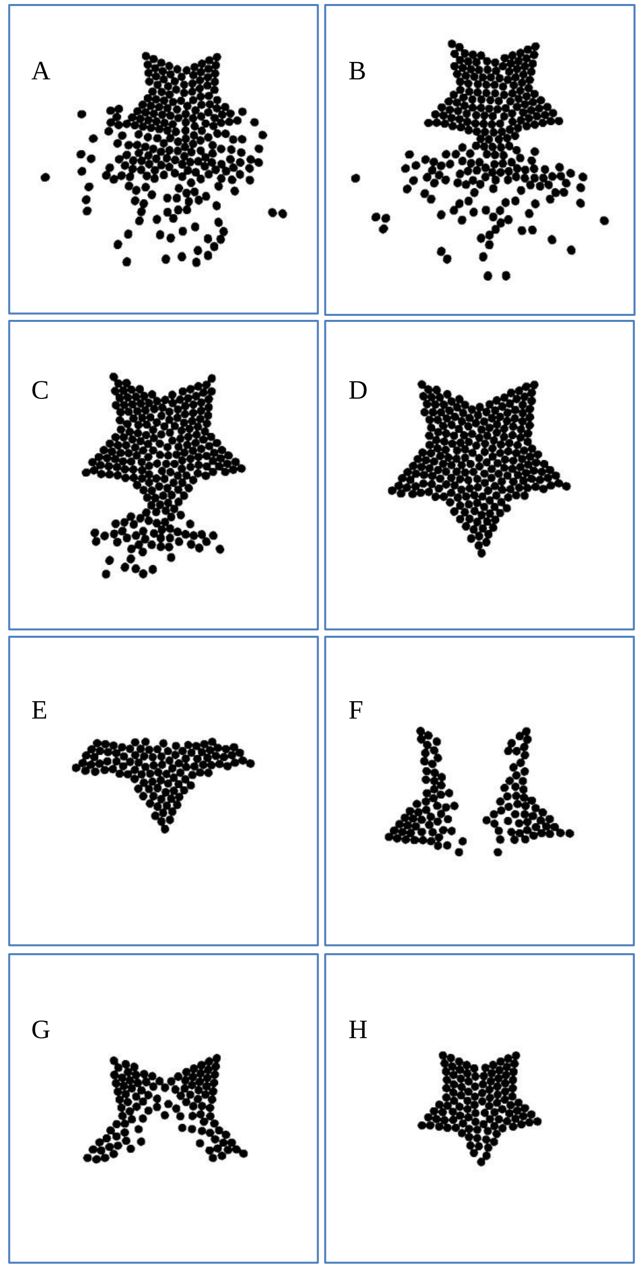

Figure 1. A demonstration of S-DASH running on a collective of robots, forming the desired shape of a star at a proportional scale. (A) An initial formation at too small a scale. (B-C) S-DASH increasing the scale. (D) The shape reaches the correct scale. (E) Half the robots are removed. (F-G) S-DASH reduces the scale. (H) The shape reaches a new correct scale.

In the previous work of S-DASH, the collective behaviors are demonstrated only in simulation. These simulated robots are very simple in their capabilities. They are capable of

Manuscript received April 4, 2010.

Michael Rubenstein is a postdoctoral fellow at Harvard University, 33 Oxford Street room 253, Cambridge, MA 02138. mrubenst@seas.harvard.edu

Radhika Nagpal is an Associate Professor of Computer Science at the School of Engineering and Applied Sciences & Wyss Institute for Bioinspired Engineering, Harvard University. 33 Oxford Street room 235, Cambridge, MA 02138. rad@eecs.harvard.edu

moving forward, rotating, communicating with local neighbors, and measuring the distance between themselves and their local neighbors. Many robot platforms exist that have these capabilities, for example [2,5,6], and therefore in theory are capable of demonstrating S-DASH. However, these robots are not practical to operate as a collective on the order of 2^{10} robots.

For an example of why they are not practical in such large numbers, consider the simple task of powering on the robots in the collective. With a standard robot, for example the E-PUCK [6] turning on the robot requires a user to toggle a power switch located on the side of the individual robot. If one robot could be powered up this way on average every 2 seconds, it would still take a single user over 30 minutes to power on all 1024 robots in the collective! Some other reasons these robots are not practical for such a large scale collective include: robot cost, operability (powering, charging, programming, etc...), and physical size.

This abstract will discuss our work-in-progress at creating Kilobot, a robot module designed to easily operate in collective sizes of 2^{10} robots or greater. In section 2, the Kilobot hardware is reviewed, including the locomotion, communication, control and power systems. In section 3 Kilobot operations are discussed which include: charging methods, programming, and power control. In section 4, some demonstrations of Kilobot's current capabilities will be given. The abstract will then conclude with a discussion on what work is left to operate S-DASH on 2^{10} Kilobots so that they demonstrate self-assembly and self healing.

II. Kilobot Hardware

The Kilobot module is designed to meet the requirements of S-DASH, while at the same time easily operate in large collectives (more than 2^{10} robots). The requirements of S-DASH are that the robot be able to move forward, turn, communicate with neighbors, and measure distances to neighbors. Kilobot meets these requirements, while keeping the design balanced against other needs for operating a large collective, such as keeping the cost per robot under $15(US), and ease in programming. Fig. 2 shows a prototype version of Kilobot.

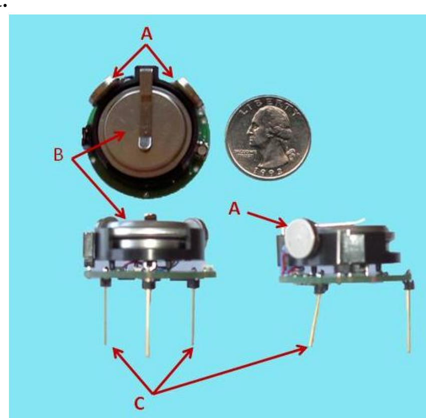

Figure 2. A prototype Kilobot module (top view in upper left, front view in lower left, side view in lower right) next to a US quarter for scale. Some features are (A) vibration motors, (B) battery, (C) slanted legs.

A. Locomotion

Kilobots uses two vibration motors along with passive directional legs, a method inspired by [7], for simple, low cost locomotion. Fig. 3. demonstrates the principal of this vibration based locomotion. In Fig. 3a, a vibration motor (black) is mounted on a green chassis. The platform is supported above the ground by two slanted legs attached to the chassis. When the motor vibrates, it causes forces that move the platform up and down, as shown in Fig. 3b. That up and down movement is translated to forward movement from the slanted shape of the legs, as shown in Fig. 3c.

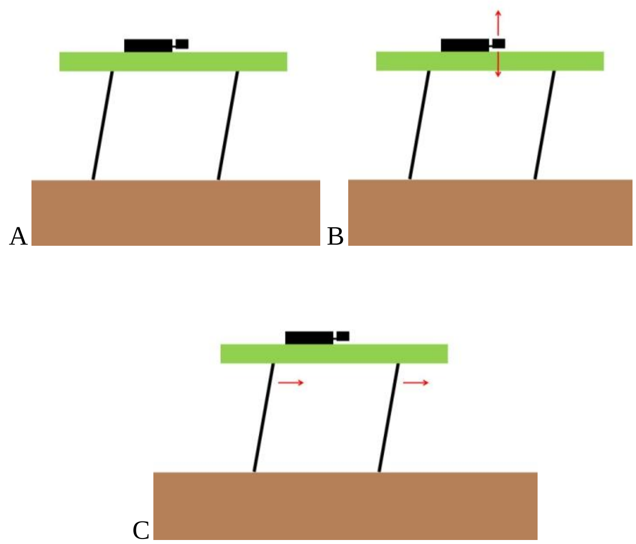

Figure 3. Vibration motion example, a side view, with gravity downward. (A) A vibration motor (black) mounted on top of a chassis (green) supported off the ground (brown) by two slanted legs. (B) The motion caused by vibration. (C) The motion of the chassis caused by the vibration of the slanted legs.

To allow turning and forward movement of the Kilobot, two vibration motors are mounted on the chassis, one on the left side, and one on the right, as shown in Fig. 4. Underneath each vibration motor is one directional leg, slanted to cause a forward force when the motor above it is vibrating. Due to the motors and legs being offset from the center of the chassis, the forward force generated when a motor is vibrating will cause the chassis to rotate. For example, if the left motor is activated, and the right motor is off, the chassis will rotate clockwise as viewed from above. If both motors are activated equally, then the chassis will move approximately forward.

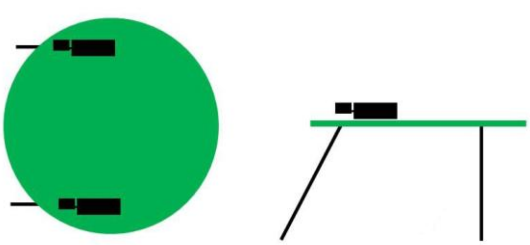

Figure 4. Kilobot locomotion design (left) top view, (right) side view.

There are three reasons for choosing this form of locomotion over traditional wheeled locomotion. The first reason is cost. Due to the complexity of the wheeled drive train, they are at least 10 times the cost of the vibration system used for Kilobot. The second reason is that the structure for a wheeled system is generally a few times larger than the vibration system used. Thirdly, the vibration system allows for minimal structure between the bottom of the chassis and the floor below. The only structures in that space are the two slanted legs on the sides, and a third straight leg in the front used to complete the tripod of legs. The reason minimal structure in this space is important will be explained in the next section.

B. Communication

The communication system on each robot serves two purposes: to send digital information to neighboring robots, and to measure the distance between itself and a neighboring robot. This is accomplished using a wide field-of-view infra-red emitter and transmitter that are located at the center of the chassis underside. The emitter and transmitter are pointed down at the floor below. As shown in Fig. 5, the infra-red light reflects off the floor below, and then is received by a neighboring robot. This communication channel can be used to send digital information and estimate the distance of that communication using the received signal strength. This communication path is the reason that it is important to minimize the structure between the chassis and the floor, because a large structure would block some or all of the communication path.

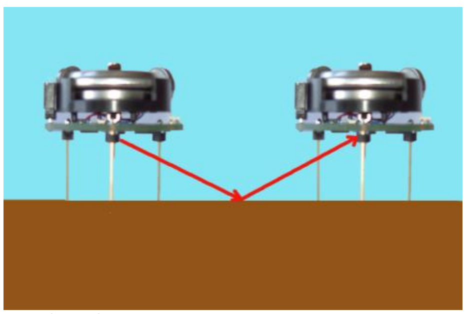

Figure 5. A path for infra-red communication between two robots.

C. Controller

Each Kilobot has a microcontroller onboard to handle the entire control of the robot, including communication, motor control and decision making. The microcontroller used is a 32 pin Atmega 168, which can be purchased in bulk for under $3 (US). This microcontroller has 16 KB of memory.

D. Power System

Each Kilobot is powered by a 160 milliamp hour, 3.4 volt lithium ion battery. The Atmega microcontroller is directly connected to the battery, while the rest of the circuitry is connected to a 3V switchable regulator. This regulator can be turned on and off by the microcontroller. Under normal usage the battery should provide approximately 4 hours of operational power for the robot. There is also a battery charging circuit on each robot, which will charge the battery if 6 volts is applied to the two slanted legs of the robot.

III. Operations

There are some operations for a standard robot that are simple, and hardly considered when designing that robot, for example: how to turn on the robot, how to charge its battery, and how to program it. These simple operations must be carefully considered when dealing with a collective of thousands of robots, to prevent them from becoming expensive, tedious, or time consuming.

A. Power Control

As demonstrated before, even a simple power switch for each robot is too time consuming for a large collective. Kilobots are designed so that the entire collective can be turned on or off in under one minute. Instead of completely disconnecting the battery from the robot when it is not in use, the robot just enters a low power state when commanded to by the user. When the user wants to activate the robots, they are sent a wake up message, and the robots will exit the low power state, entering the operational mode.

This is done by having the microcontroller turn off the switchable power regulator, and then entering a deep sleep mode for 1 minute when the robot is not in use. While Kilobot is in this deep sleep mode, the onboard circuitry is only drawing 50μA of current from the battery. After about a minute of this deep sleep, the microcontroller will wake up, turn on the switchable power regulator, and for 10ms, check to see if it detects a wake up message over the infra red communication channel. If this message is detected, then the robot will enter the operational mode. If the message is not detected, the Kilobot will re-enter the deep sleep mode for another minute, repeating the cycle. The experimenter uses an infra-red remote control to send messages to all the robots in the collective to either enter the deep sleep cycle, or to wake up from the deep sleep cycle.

B. Charging

As stated earlier, each robot has a built in lithium ion battery charger, that will charge the on board battery when 6 volts is applied to the two slanted legs. The charger will automatically cease charging when the battery becomes full. This allows bulk charging of the Kilobots by placing each robot on a set of conducting strips attached to a 6v power supply, as visualized in Fig 6. This charging method is inspired from [8].

Figure 6. A visualization of the Kilobot charging scheme.

C. Programming

The standard way to program a robot is by connecting it to a computer and then programming it, which is too time consuming for a large collective. Two possible methods for programming a Kilobot speed up the programming process by taking advantage of the Atmega's self-programmable memory. The first way would be to program one Kilobot the traditional way, using a cable connected to a computer. Next, place this newly programmed Kilobot into a group of robots you wish to program. This Kilobot will then broadcast over the infra red communication channel its new program to all of its surrounding neighbors, who will overwrite their old program with the newer program. These neighbors will then propagate the new program to all of their neighbors with the older program. This process will continue until all the robots in the group have the new program. A second possible method to program a group of Kilobots is to use a powerful infra-red light that can broadcast the new program to all the robots at once. When the robots receive the program from the infra-red light, they again use the self-programmable memory feature of the Atmega and replace their old program with the new one.

IV. Current Capabilities

The development of the Kilobot system is currently still in progress, so not all of the goals of the Kilobot system have been achieved yet. This section will describe what capabilities are needed for the Kilobot to achieve the collective self-assembling and self-healing behaviors of S-DASH. Furthermore, it will discuss which ones have already been achieved, and which ones are still incomplete.

A. Movement

The robots must have the capability to move to any location in a 2D plane. This capability has already been achieved using vibration based locomotion. Using this technique, a Kilobot is capable of turning in a circle of less than 3 robot radii, and moving at speeds greater than 2 robot radii per second.

B. Power Control

The ability to put the robots into a deep sleep, and then wake them up to the operational mode is still incomplete. The circuitry for the deep sleep has been built and tested, however the full functionality has not yet been programmed into the Kilobots.

C. Programming

The ability to program the robots as a group is incomplete. Again, the circuitry is built and tested, but this capability has not been fully programmed yet.

D. Charging

The large scale charging system for the Kilobot has been fully tested, and is operating as described.

E. Communication

The communication system on the Kilobot has been built and fully tested. Each robot is capable of sending a four byte data packet to its neighbors up to 750 times a second. A Kilobot can communicate to any neighbor that is closer than 8 robot radii away.

F. Distance Sensing

Every time a communication packet is received by a Kilobot it also has the ability to measure the distance between itself and the transmitting robot. This distance measurement has the accuracy of ±2mm.

G. Triangulation

Kilobot has demonstrated the capability to triangulate its location in the environment. The triangulation works as follows: given three neighboring robots in a triangle formation, where each robot knows its location in the environment, a forth robot can then determine its position in the environment. This fourth robot does so after receiving a data packet from each neighbor that gives that neighbor's location in the environment. That communication also allows the fourth robot to measure the distance from itself to each of its neighbors. From this information, the fourth robot can determine its location.

H. Trilateration

Using trilateration, a method described in [9,10], each robot should be able to assign itself a unique location in a coordinate system. This coordinate system is agreed upon by all the other Kilobots in the collective. The unique location for each robot can be developed solely from communication between neighboring robots, and the measuring of that distance. While the hardware is ready for trilateration, the actual software is still in production, and has not been fully tested.

I. Demonstrations

There are two demonstrations that can be presented at the workshop either by video, or bringing the robots. In the first demonstration, one Kilobot is stationary, and another orbits around the first at a fixed distance. This demonstrates two capabilities: first, that distance between Kilobots can be reliably sensed, and second, that the method used for locomotion is reliable and accurate when feedback from the environment is available.

A second demonstration shows triangulation in action. In this demonstration, three Kilobots are placed in a triangle. A fourth robot is moved about in the environment, and using a RGB LED, it indicates its location in the coordinate system. This demonstration shows both communication and distance sensing.

Conclusion

This abstract presents the Kilobot robot, designed specifically for operation in a large collective. While the Kilobot is a simple low cost robot, it is expected to be able to self-assemble and self-heal its collective shape. Currently there are just a few Kilobots produced for early prototyping and testing, however once the design is finalized, we plan to produce 1024 robots.

REFERENCES

[1] Sahin, E, et. al. "SWARM-BOT: Pattern Formation in a Swarm of Self Assembling Mobile Robots." International Conference on Systems, Man and Cybernetics. Hammamet, Tunisia, 2002. 145-150.

[2] Chiu, H., Rubenstein, M., Shen, W-M. "Deformable Wheel'-A Self-Recovering Modular Rolling Track." Intl. Symposium on Distributed Robotic Systems. Tsukuba, Japan, 2008.

[3] Rubenstein, M., Shen, W-M. "Scalable Self-Assembly and Self-Repair In A Collective Of Robots" IROS, Oct 2009.

[4] Rubenstein, M., Shen, W-M. "Automatic Scalable Size Selection for the Shape of a Distributed Robotic Collective" IROS, Oct 2010.(submitted)

[5] P. Maxim, S. Hettiarachchi, W. Spears, D. Spears, J. Hamman, T. Kunkel, C. Speiser. *Trilateration localization for multi-robot teams.* Sixth International Conference on Informatics in Control, Automation and Robotics, Special Session on Multi-Agent Robotic Systems. 2008.

[6] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D. and Martinoli, A. (2009) The e-puck, a Robot Designed for Education in Engineering. Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, 1(1) pp. 59-65.

[7] http://www.evilmadscientist.com/article.php/bristlebot

[8] S. Kornienko, O. Kornienko, A. Nagarathinam, and P. Levi. From real robot swarm to evolutionary multi-robot organism. In Proc. of the CEC2007, Singapore, 2007.

[9] I. Borg, *Modern Multidimensional Scaling: Theory and Applications.* Springer Series in Statistics. 2005.

[10] Moore, D. Et. al. "Robust Distributed Network Localization with Noisy Range Measurements". Sensys 2004.